

Object Oriented
Programming
in Java



»Лекционен курс

»ООП1 (Java)



Базови езикови
конструкции >

1



Въвеждащ Пример



Какво прави програмата?

```
import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Здравейте!");
        System.out.println("Ще събира две числа за Вас.");
        System.out.println("Въведете две цели числа на един ред:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("Сумата на двете числа е");
        System.out.println(n1 + n2);
    }
}
```

Събира две цели числа

Изпълнение на програмата



Какъв е резултатът?

```
import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Здравейте!");
        System.out.println("Ще събера две числа за Вас.");
        System.out.println("Въведете две цели числа на един ред:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("Сумата на двете числа е");
        System.out.println(n1 + n2);
    }
}
```

Здравейте!

Ще събера две числа за Вас.

Въведете две цели числа на един ред:

2 3

Сумата на двете числа е

5

Потребител

- » Потребител: човекът, който взаимодейства с програма
- » В определени случаи потребителят може да бъде програмистът, т.е. човекът, който е написал програмата
 - > Обикновено студентите често са програмисти и потребители
- » В разработването на реален софтуер програмистът и потребителят обикновено са различни хора

Въвеждащ пример



Каква е програмата?

```
import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Здравейте!");
        System.out.println("Ще събера две числа за Вас.");
        System.out.println("Въведете две цели числа на един ред:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("Сумата на двете числа е");
        System.out.println(n1 + n2);
    }
}
```

Интерактивна

Структура на програмата

```
import java.util.Scanner;  
  
public class FirstProgram {  
    public static void main(String[] args) {  
        System.out.println("Здравейте!");  
        System.out.println("Ще събера две числа за вас.");  
        System.out.println("Въведете две цели числа на един ред:");  
  
        int n1, n2;  
  
        Scanner keyboard = new Scanner(System.in);  
        n1 = keyboard.nextInt();  
        n2 = keyboard.nextInt();  
        System.out.println("Сумата на двете числа е");  
        System.out.println(n1 + n2);  
    }  
}
```

Получава класа Scanner от пакета (библиотека) java.util

Име на класа – по ваши избор. Тази програма трябва да е във файл с име FirstProgram.java

Изпраща изход на екран

Казва, че n1 и n2 са променливи, които съдържат цели числа (int)

Прочита програмата за въвеждане от клавиатурата

Чете цяло число от клавиатурата

Обобщение

```
import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Здравейте!");
        System.out.println("Ще събера две числа за Вас.");
        System.out.println("Въведете две цели числа на един ред:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("Сумата на двете числа е");
        System.out.println(n1 + n2);
    }
}
```

Не се притеснявайте, ако
някои от детайлите на
програмата не са напълно
ясно за това първо четене.

Императивно програмиране

» Алгоритъм:

- > Подход за изчисляване на търсene стойности от дадени такива, . . . който използва постъпково изпълнение на елементарни обработващи операции

» Императивна програма:

- > Алгоритми, описани посредством обръщения (достъп) към стойности на променливи
- > Четене и промяна на стойности

*Императивно програмиране е ориентирано
към описание на алгоритми*

Императивна програма



Основни части на една императивна програма?

```
import java.util.Scanner;

public class FirstProgram
{
    public static void main(String[] args)
    {
        System.out.println("Здравейте!");
        System.out.println("Ще събира две числа за Вас.");
        System.out.println("Въведете две цели числа на един ред:");

        int n1, n2;

        Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt(); ← Получава входни данни

        System.out.println("Сумата на двете числа е");
        System.out.println(n1 + n2);
    }
}
```

Извежда резултати

Получава входни данни

Обработка данни

Императивно програмиране

» Базови концепции:

- > **Променливи:** притежават стойности, които се променят посредством прилагане на оператори
- > **Оператори:** Служат за достъп до стойностите на променливите (четене, промяна, извеждане на стойностите)

» Базов метод за структуриране на императивното програмиране:

- > **Процедури (функции, методи):** Частични алгоритми, представени като оператори на езика

Императивно програмиране

- » Данни (променливи) и обработващи алгоритми (процедура, функция, метод) **са разделени** структури в императивното програмиране
- » За сравнение с обектно-ориентираното:
 - > Клас = единство от данни и обработващи алгоритми

Java: основно предназначение обектно-ориентирано програмиране.
Одаче, императивното програмиране е възможно.

Компоненти



От колко компонента (класа) се състои програмата?

```
import java.util.Scanner;

public class FirstProgram 3
{
    public static void main(String[] args)
    {
        System.out.println("Здравейте!");
        2 System.out.println("Ще събера две числа за Вас.");
        System.out.println("Въведете две цели числа на един ред:");

        int n1, n2;

        1 Scanner keyboard = new Scanner(System.in);
        n1 = keyboard.nextInt();
        n2 = keyboard.nextInt();

        System.out.println("Сумата на двете числа е");
        System.out.println(n1 + n2);
    }
}
```

3

Компоненти на програмата



Зашо възможно?

Разделено компилиране

Дефиниран от потребителя
клас

```
class FirstProgram {  
    ...  
}  
File: FirstProgram.java
```

```
class Scanner {  
    ...  
}  
File: Scanner.java
```

```
class System {  
    ...  
}
```

Java API(Application Programming Interface)
(стандартни библиотеки)

Интерфейси между компонентите

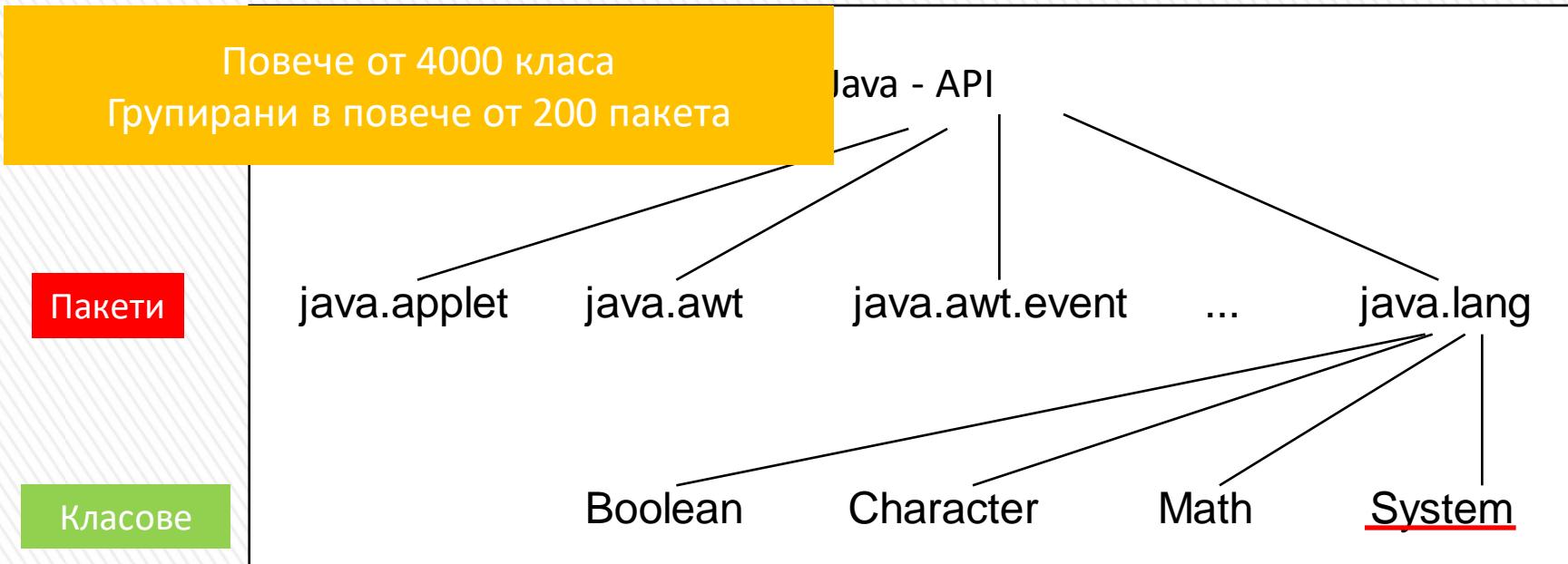
```
class FirstProgram {  
  
    public static void main (String [] args) {  
        ...  
        keyboard = Scanner...;  
  
        System.out.print(" deg F is ");  
        ...  
    }  
}
```

Java API

```
class System {  
    ...  
}
```

```
class Scanner {  
    ...  
}
```

Организация на Java-API



Пакети:
Сбирка от класове

Класове:
Софтуерни компоненти

Име на пакет (напр. java.awt.event) отразява Directory-Names:
→ /java.awt/event

Спецификация на пакети

Java™ Platform, Standard Edition 7 API Specification

This document is the API specification for the Java™ Platform, Standard Edition.

See: Description

Packages	Description
java.applet	Provides the classes necessary to create applets.
java.awt	Contains all of the classes for creating graphical user interfaces.
java.awt.color	Provides classes for color spaces.
java.awt.datatransfer	Provides interfaces and classes for transferring data between applications.
java.awt.dnd	Drag and Drop is a direct manipulation element in the GUI.
java.awt.event	Provides interfaces and classes for defining event-handling code.
java.awt.font	Provides classes and interface relating to font rendering.
java.awt.geom	Provides the Java 2D classes for defining geometric shapes.
java.awt.im	Provides classes and interfaces for the input method framework.
java.awt.im.spi	Provides interfaces that enable the development of input methods that can be used with any Java runtime environment.
java.awt.image	Provides classes for creating and modifying images.
java.awt.image.renderable	Provides classes and interfaces for producing rendering-independent images.
java.awt.print	Provides classes and interfaces for a general printing API.
java.beans	Contains classes related to developing beans -- components based on the JavaBeans™ architecture.
java.beans.beancontext	Provides classes and interfaces relating to bean context.
java.io	Provides for system input and output through data streams, serialization and the file system.
java.lang	Provides classes that are fundamental to the design of the Java programming language.
java.lang.annotation	Provides library support for the Java programming language annotation facility.
java.lang.instrument	Provides services that allow Java programming language agents to instrument programs running on the JVM.
java.lang.invoke	The java.lang.invoke package contains dynamic language support provided directly by the Java core class libraries and virtual machine.
java.lang.management	Provides the management interfaces for monitoring and management of the Java virtual machine and other components in the Java runtime.
java.lang.ref	Provides reference-object classes, which support a limited degree of interaction with the garbage collector.

Пакет: java.lang

- Автоматично се доставя от системата
- Не е необходимо да се импортира

Java програми

- » Принципно два вида Java програми:
 - > Приложения
 - > Аплети
- » Могат да съдържат повече от един клас
- » При приложението точно един клас трябва да съдържа дефиниция на метод с име **main**
 - > Активира се когато започне изпълнението на приложението

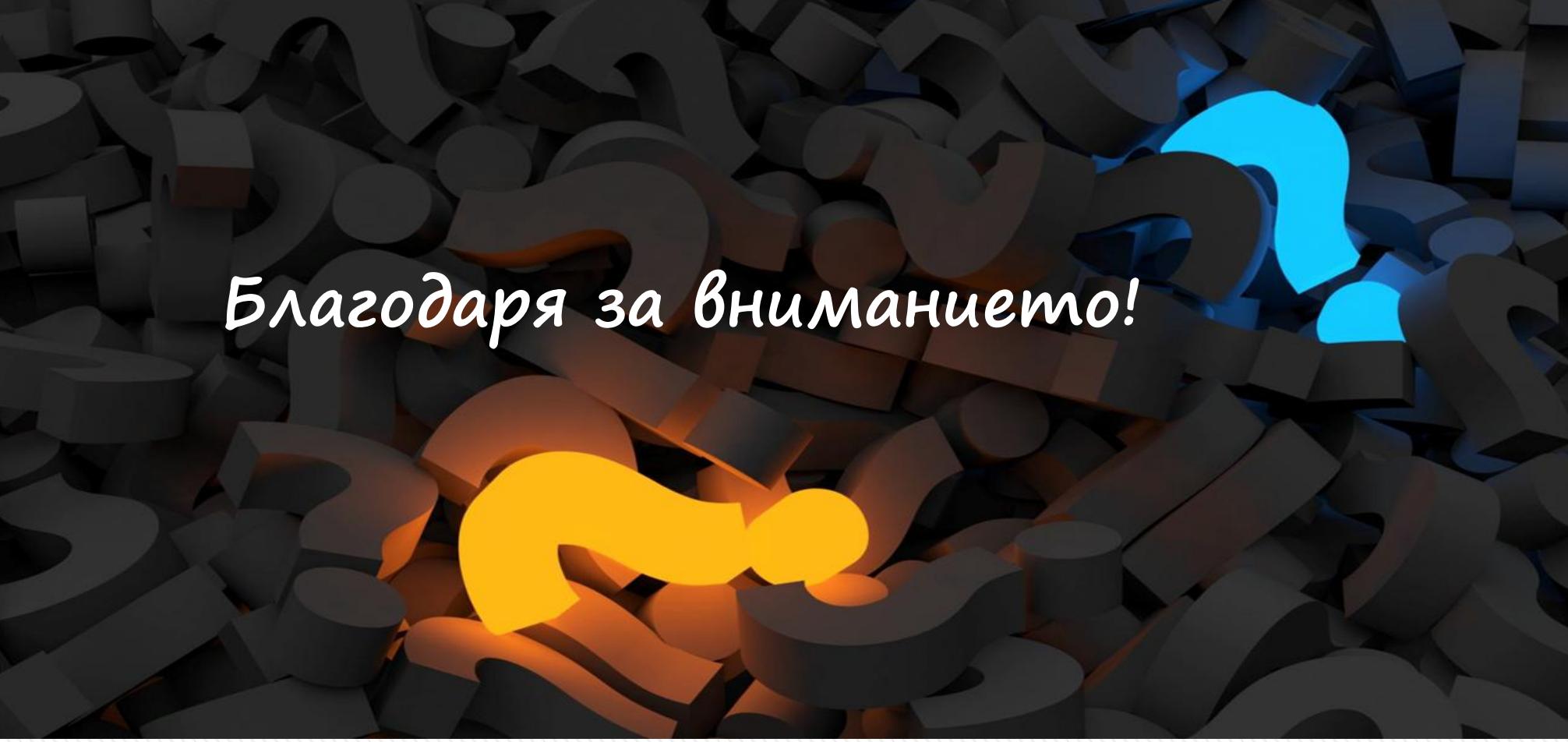
Структура на Java-програми

Клас = базов компонент

```
Име клас           Тяло клас (начало)  
class FirstProgram {  
    public static void main (String [] args) {  
        int n1,n2;  
        ...  
    } }  
Декларация метод  
Декларация променлива
```

Клас: 'Множество' от
- Декларации на променливи
- Декларации на методи
(метод = алгоритъм, процедура, функция)

За общата програма: един метод **main()** → там започва обработката !



Благодаря за внимание!